Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Fall, Year:2023), B.Sc. in CSE (Day)

Predicting Online Food Ordering Purchase

Course title: Data Mining Lab
Course Code: CSE 436 **Section:** 201 D5

<u>Students Details</u>

Name	ID
A.K.M. Atiqur Rahman	202002035

Submission Date : 14-01-2024
Course Teacher's Name: Mr. Mozdaher Abdul Quader

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Introduction

The project, titled "Predicting Online Food Ordering Purchase" explores the use of machine learning techniques such as logistic regression or random forest to forecast food orders placed via online delivery platforms. With the advent of FoodPanda, HungryNaki, and Grab, the demand for online food deliveries has soared. To streamline delivery processes, companies harness customer data to predict future orders and optimize delivery routes. This project aims to demonstrate how classification algorithms, such as logistic regression or random forest, can predict customer behavior regarding food orders. By leveraging Python and these algorithms, the goal is to showcase their potential in improving delivery efficiency and aiding strategic business decisions in the food delivery industry.

1.2 Motivation

The motivation driving the exploration of Predicting Online Food Ordering Purchase lies in several key aspects:

- The primary goal is to optimize food delivery by anticipating customer orders, ensuring timely service and customer satisfaction.
- By predicting customer behavior, companies can identify areas and customer segments with higher order probabilities, aiding in resource allocation and business expansion.
- Utilizing the vast customer data available, companies can forecast re-order probabilities, enabling proactive measures to retain customers.
- Applying predictive models in the food delivery sector showcases the practical utility of machine learning in optimizing business operations.

The core motivation behind this project is to harness predictive models to enhance the efficiency of food delivery services, leveraging customer data to streamline operations and improve customer experience.

1.3 Problem Definition

1.3.1 Problem Statement

The problem statement for the project "Predicting Online Food Ordering Purchase" encompasses the following inquiries:

- How can machine learning algorithms effectively used for data mining project which is to predict customer behavior regarding online food orders to enhance delivery efficiency and customer satisfaction?
- What approach can be devised to leverage historical customer data for predicting future online food orders, facilitating proactive measures for delivery service optimization?
- How can predictive models, such as Logistic Regression and Random Forest, be applied and compared to predict customer re-order probabilities accurately?

By addressing these problem statements, this project endeavors to develop predictive models that leverage machine learning and Data Mining algorithms to forecast online food orders, thereby aiding delivery companies in optimizing their services and improving customer experiences.

1.3.2 Complex Engineering Problem

The predictive analysis of online food ordering purchase using data mining techniques encompasses several intricate engineering challenges essential for a successful implementation. These complexities are summarized below:

Attribute	How to Address and Relevance to Project
Algorithmic Expertise:	Acquiring expertise in various data mining methodologies requires continual learning and collaborative knowledge sharing within the team, crucial for the project's success.
Performance Optimization:	Implementing efficient algorithms and computational strategies is pivotal to enhancing model performance, ensuring scalability, and real-time analysis, aligning directly with the project's objectives.
Data Processing Depth:	Employing robust data preprocessing techniques and scalable infrastructure for comprehensive data analysis and pattern recognition is highly pertinent to achieving the project's goals.
Stakeholder Engagement:	Maintaining continuous engagement with stakeholders, aligning data mining choices with their expectations is critical for effective utilization and project success.
System Integration:	Ensuring seamless integration of various data mining techniques into a unified system enhances usability and practical application, directly impacting the project's objectives.

Table 1.1: Summary of Attributes and Their Relevance

Addressing these complex engineering challenges demands a multidisciplinary approach encompassing algorithmic expertise, performance optimization, comprehensive data processing, stakeholder engagement, and seamless system integration. The successful execution of this project aims to empower practitioners in making informed decisions while leveraging data mining techniques for predicting online food ordering purchases.

1.4 Design Goals/Objectives

In this project, the objectives are:

- To explore and analyze various data mining techniques in order to predict and optimize online food ordering purchase patterns.
- To compare the effectiveness and suitability of different data mining methodologies in accurately forecasting customer behavior in online food ordering.
- To develop a predictive system that assists food delivery companies in optimizing delivery processes and resource allocation based on anticipated order volumes and timings.

1.5 Application

The project on predicting online food ordering purchase holds significant applications across various domains:

- Food Delivery Industry: Assisting food delivery companies in optimizing delivery operations by predicting customer orders, leading to efficient resource allocation and faster delivery.
- Customer Relationship Management: Enhancing customer retention strategies by predicting customer ordering behaviors, allowing personalized marketing campaigns and loyalty programs.
- Market Analysis: Aiding market analysts and businesses to understand food consumption trends, popular food items, and customer preferences in different demographics.
- Resource Planning: Enabling restaurants and delivery services to manage inventory, staff, and supplies based on anticipated order volumes and timings.

The practical implications of the "Predicting Online Food Ordering Purchase" project extend to revolutionizing the food industry, offering insights that can improve operational efficiency, customer satisfaction, and strategic decision-making for food delivery services and related businesses.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

The evolution of online food delivery services catalyzed a substantial shift in consumer habits, propelled by platforms like FoodPanda, HungryNaki, and Grab. This transformation has prompted a surge in demand for predictive models that can anticipate future food orders. The aim of this project is to employ robust data mining techniques, specifically classification algorithms such as Logistic Regression or Random Forest, to predict whether customers will engage in subsequent food orders. By harnessing these algorithms, the objective is to streamline delivery operations, optimize user experiences, and facilitate targeted marketing strategies for food delivery enterprises. This project delves into the task of online food order prediction, drawing insights from customer purchasing behavior. Leveraging machine learning methods, the focus lies in predicting the likelihood of a customer placing another food order based on their historical buying patterns. Through the implementation of classification algorithms, the endeavor is to enhance operational efficiencies within the food delivery ecosystem and enable companies to proactively address customer preferences, ultimately improving service quality and customer satisfaction.

2.2 Key Components

In the predicting online food ordering purchases, several crucial components contribute to the efficacy of the analysis:

- **Feature Selection:** Identifying and selecting pertinent features such as customer demographics, order history, location, and feedback are crucial. These features play a pivotal role in predicting future ordering behavior.

- **Data Preprocessing:** Cleaning, transforming, and normalizing the data is essential. Ensuring data consistency and quality enhances the accuracy of predictive models.

- **Model Selection and Training:** Choosing appropriate predictive models, such as logistic regression or decision trees, and training them using historical data is key. The models should accurately capture patterns in the data.
- **Evaluation Metrics:** Metrics like accuracy, precision, recall, and F1-score help measure model performance. These metrics quantify the predictive power of the models.
- **Deployment and Monitoring:** Deploying the trained models into a production environment for real-time predictions is vital. Continuous monitoring and recalibration ensure the models remain accurate over time.

These components collectively form the foundational elements for predicting online food ordering purchases, enabling the development of accurate and reliable predictive models.

2.3 Tools and Libraries

Here are some tools and libraries used in the project:

- **Software Tools:**
  - Desktop/Laptop
  - Google Colab
  - Microsoft Edge
- **Python Libraries & Techniques:**
  - Classification Techniques: Logistic Regression or Random Forest.
  - Pandas: For data manipulation and analysis
  - Scikit-learn: For machine learning algorithms
  - Matplotlib: For data visualization
  - Seaborn: For statistical data visualization
  - Plotly: For interactive and web-based visualizations
  - Ipywidgets: For interactive widgets in Jupyter Notebooks
  - IPython.display: For displaying content in Jupyter Notebooks

Overall, these are the main tools and libraries that can be used to build a Predicting Online Food Ordering Purchase project.

2.4 Algorithms

Here's an algorithm outlining the steps for predicting online food ordering purchase using Data Mining techniques:

Algorithm 1: Predicting Online Food Ordering Purchase

Input: Dataset with information on customers' demographics, feedback, and ordering habits
Output: Prediction of whether a customer will order again
Data: Widgets for user input: age, gender, marital status, occupation, monthly income, education, family size, pin code, feedback

Load the dataset
Map categorical variables to numeric values
Create widgets for user input
Initialize the model for prediction
Function `predict button clicked()`:
    Retrieve user inputs from widgets
    Collect user features
    Predict using the model
    Display the prediction probabilities
    Visualize the probabilities using a bar chart
Create a button for prediction
while *Button not clicked* **do**
    if *Button clicked* **then**
        Retrieve selected values from widgets
        Predict and display the result
Display the widgets and prediction button for user interaction;

2.5 Implementation

2.5.1 Importing Libraries & Loading Data-set:

```
import numpy as np
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style="whitegrid")

data = pd.read_csv("/content/onlinefoodss.csv")
print(data.head())
```

2.5.2 Summary information about the dataset:

```
print(data.info())
```

2.5.3 Visualize online food order decisions based on the age of the customer:

```
plt.figure(figsize=(8, 6))
plt.title("Online Food Order Decisions Based on the Age
    of the Customer")
sns.histplot(x="Age", hue="Output", data=data)
plt.show()
```

2.5.4 Visualize online food order decisions based on the size of the family of the customer:

```
plt.figure(figsize=(8, 6))
plt.title("Online Food Order Decisions Based on the Size
     of the Family")
sns.histplot(x="Family size", hue="Output", data=data)
plt.show()
```

2.5.5 create a dataset of all the customers who ordered the food again:

```
buying_again_data = data.query("Output == 'Yes'")
print(buying_again_data.head())
```

2.5.6 Find who orders food more online:

```
gender = buying_again_data["Gender"].value_counts()
label = gender.index
counts = gender.values
colors = ['Red','Green']

fig = go.Figure(data=[go.Pie(labels=label, values=counts
    )])
```

```
fig.update_layout(title_text='Who Orders Food Online More: Male Vs Female')
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=30,
                  marker=dict(colors=colors, line=dict(color='black', width=3)))
fig.show()
```

2.5.7 The marital status of the customers who ordered again:

```
marital = buying_again_data["Marital Status"].value_counts()
label = marital.index
counts = marital.values
colors = ['Red','Green']

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Who Orders Food Online More: Married Vs Singles')
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=30,
marker=dict(colors=colors, line=dict(color='black', width=3)))
fig.show()
```

2.5.8 The income group of the customers who ordered the food again:

```
income = buying_again_data["Monthly Income"].value_counts()
label = income.index
counts = income.values
colors = ['Red','Green']

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Which Income Group Orders Food Online More')
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=30, marker=dict(colors=colors, line=dict(color='black', width=3)))
```

```
fig.show()
```

2.5.9 Importing necessary libraries:

```
import ipywidgets as widgets
from IPython.display import display
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import numpy as np
import pandas as pd
```

2.5.10 Data preprocessing:

```
# Mapping categorical features to numerical values
data["Gender"] = data["Gender"].map({"Male": 1, "Female"
    : 0})
data["Marital Status"] = data["Marital Status"].map({"
    Married": 2,
"Single": 1,
"Prefer not to say": 0})
data["Occupation"] = data["Occupation"].map({"Student":
    1,
"Employee": 2,
"Self Employeed": 3,
"House wife": 4})
data["Educational Qualifications"] = data["Educational
    Qualifications"].map({"Graduate": 1,
"Post Graduate": 2,
"Ph.D": 3, "School": 4,
"Uneducated": 5})
data["Monthly Income"] = data["Monthly Income"].map({"No
     Income": 0,
"25001 to 50000": 5000,
"More than 50000": 7000,
"10001 to 25000": 25000,
"Below Rs.10000": 10000})
data["Feedback"] = data["Feedback"].map({"Positive": 1,
    "Negative ": 0})
```

2.5.11 Defining input features and target variable:

```
x = np.array(data[["Age", "Gender", "Marital Status", "Occupation",
"Monthly Income", "Educational Qualifications",
"Family size", "Pin code", "Feedback"]])
y = np.array(data["Output"])
```

2.5.12 Training a machine learning model:

```
from sklearn.ensemble import RandomForestClassifier
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.10, random_state=42)
model = RandomForestClassifier()
# model = LogisticRegression()
model.fit(xtrain, ytrain)
```

2.5.13 Creating widgets for user input:

```
age_widget = widgets.IntSlider(description="Age:", min=18, max=100)
gender_widget = widgets.Dropdown(options=[("Male", 1), ("Female", 0)], description="Gender:")
marital_status_widget = widgets.Dropdown(options=[("Single", 1), ("Married", 2), ("Not Revealed", 3)], description="Marital Status:")
occupation_widget = widgets.Dropdown(options=[("Student", 1), ("Employee", 2), ("Self Employed", 3), ("Housewife", 4)], description="Occupation:")
income_widget = widgets.Dropdown(options=[("No Income", 0), ("25001 to 50000", 5000), ("More than 50000", 7000), ("10001 to 25000", 25000), ("Below Rs.10000", 10000)], description="Monthly Income:")
education_widget = widgets.Dropdown(options=[("Graduate", 1), ("Post Graduate", 2), ("Ph.D", 3), ("School", 4), ("Uneducated", 5)], description="Education:")
family_size_widget = widgets.IntSlider(description="Family Size:", min=1, max=10, value=4)
pin_code_widget = widgets.IntText(description="Pin Code:")
feedback_widget = widgets.Dropdown(options=[("Positive", 1), ("Negative", 0)], description="Feedback:")
```

2.5.14 Function to handle prediction button click:

```

def predict_button_clicked(btn):
    user_features = [
        age_widget.value,
        gender_widget.value,
        marital_status_widget.value,
        occupation_widget.value,
        income_widget.value,
        education_widget.value,
        family_size_widget.value,
        pin_code_widget.value,
        feedback_widget.value,
    ]

    # Predict using the model
    prediction = model.predict_proba([user_features])

    # Display prediction
    print("Predicted output (Will the customer order
       again?):", prediction)

    # Create a bar chart to visualize the probabilities
    classes = ['Negative', 'Positive']
    plt.bar(classes, prediction[0])
    plt.xlabel('Classes')
    plt.ylabel('Probability')
    plt.title('Prediction Probabilities')
    plt.show()
    print(model.score(xtest, ytest))
```

2.5.15 Creating a button widget:

```
predict_button = widgets.Button(description='Predict')
```

2.5.16 Connecting the button to the prediction function:

```
predict_button.on_click(predict_button_clicked)
```

2.5.17 Displaying the button along with the existing widgets:

```
display(age_widget, gender_widget, marital_status_widget, occupation_widget, income_widget,
        education_widget, family_size_widget, pin_code_widget, feedback_widget, predict_button)
```

Chapter 3

Performance Evaluation

3.1 Results Analysis/Testing

3.1.1 UI/UX

After running the code it will look like this:

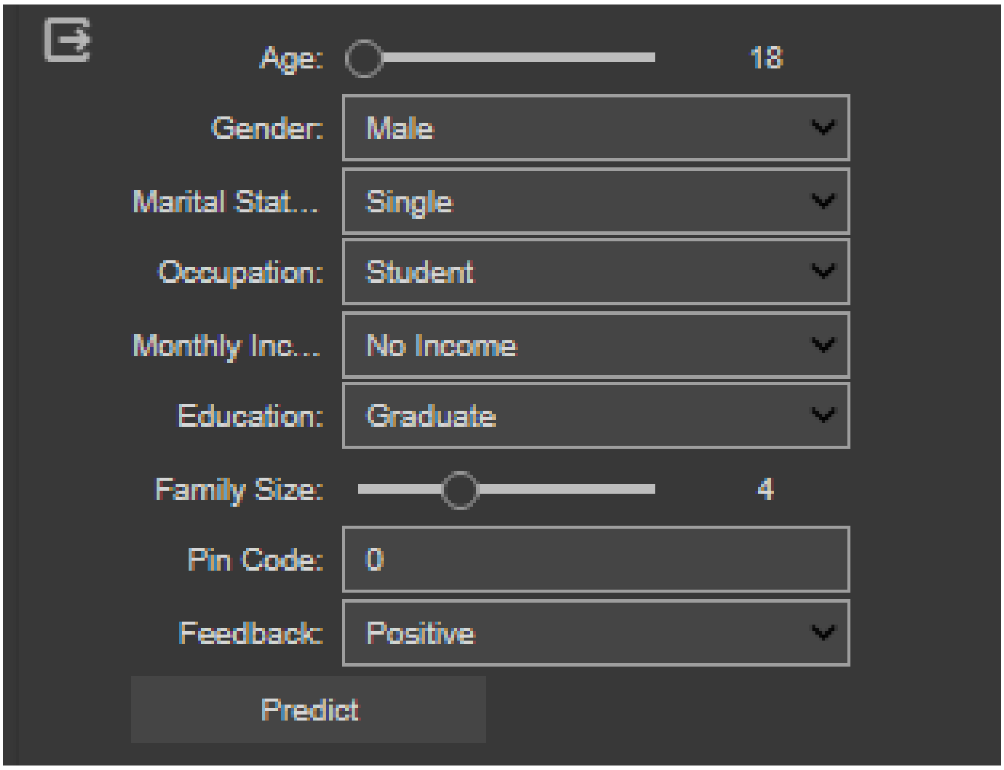

Figure 3.1: Interactive widgets

3.1.2 Final Testing

Here are some output screenshots from our project:

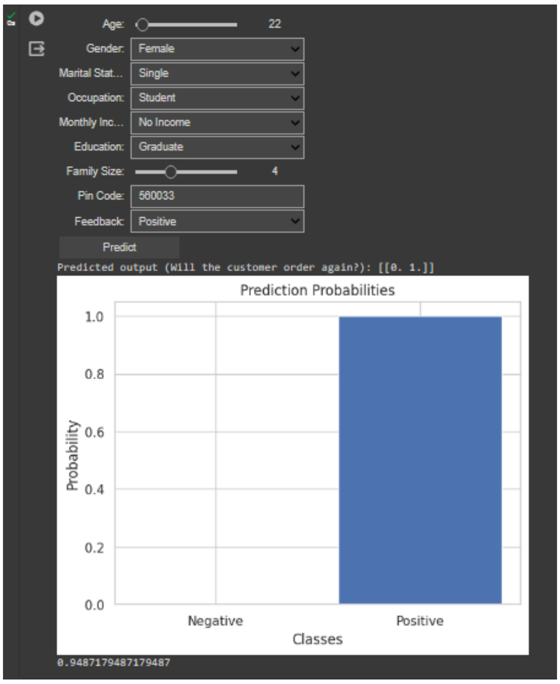

Figure 3.2: predict online food orders as Yes.

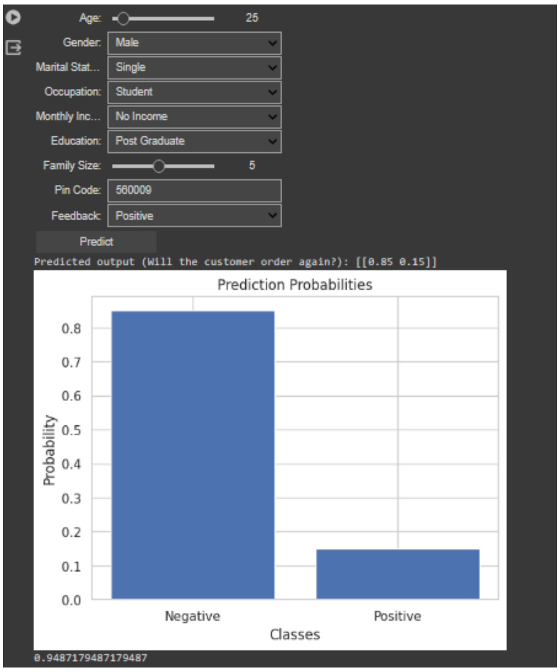

Figure 3.3: predict online food orders as No.

```
   Age  Gender Marital Status Occupation  Monthly Income  \
0   20  Female         Single    Student       No Income
1   24  Female         Single    Student  Below Rs.10000
2   22    Male         Single    Student  Below Rs.10000
3   22  Female         Single    Student       No Income
4   22    Male         Single    Student  Below Rs.10000

  Educational Qualifications  Family size  latitude  longitude  Pin code  \
0              Post Graduate            4   12.9766    77.5993    560001
1                   Graduate            3   12.9770    77.5773    560009
2              Post Graduate            3   12.9551    77.6593    560017
3                   Graduate            6   12.9473    77.5616    560019
4              Post Graduate            4   12.9850    77.5533    560010

  Output   Feedback Unnamed: 12
0    Yes   Positive         Yes
1    Yes   Positive         Yes
2    Yes  Negative          Yes
3    Yes   Positive         Yes
4    Yes   Positive         Yes
```

Figure 3.4: Display the 5 rows of the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 388 entries, 0 to 387
Data columns (total 13 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Age                         388 non-null    int64
 1   Gender                      388 non-null    object
 2   Marital Status              388 non-null    object
 3   Occupation                  388 non-null    object
 4   Monthly Income              388 non-null    object
 5   Educational Qualifications  388 non-null    object
 6   Family size                 388 non-null    int64
 7   latitude                    388 non-null    float64
 8   longitude                   388 non-null    float64
 9   Pin code                    388 non-null    int64
 10  Output                      388 non-null    object
 11  Feedback                    388 non-null    object
 12  Unnamed: 12                 388 non-null    object
dtypes: float64(2), int64(3), object(8)
memory usage: 39.5+ KB
None
```

Figure 3.5: Summary information about the dataset.

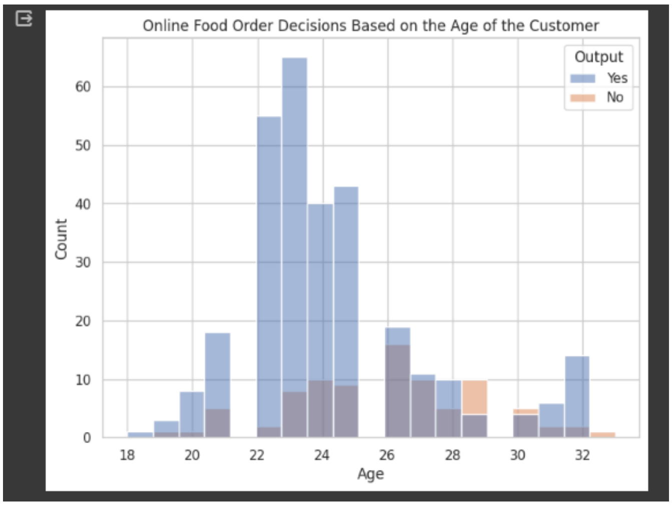

Figure 3.6: Visualize online food order decisions based on the age of the customer.

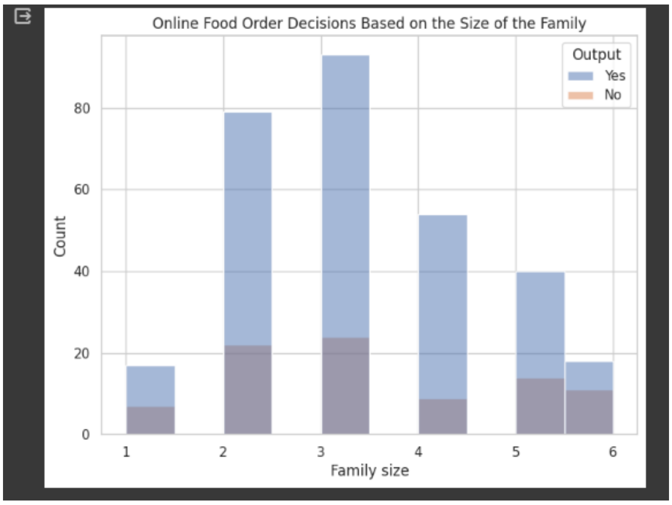

Figure 3.7: Visualize online food order decisions based on the size of the family of the customer.

	Age	Gender	Marital Status	Occupation	Monthly Income
0	20	0	1	1	0
1	24	0	1	1	10000
2	22	1	1	1	10000
3	22	0	1	1	0
4	22	1	1	1	10000

	Educational Qualifications	Family size	latitude	longitude	Pin code
0	2	4	12.9766	77.5993	560001
1	1	3	12.9770	77.5773	560009
2	1	2	12.9551	77.6593	560017
3	1	6	12.9473	77.5616	560019
4	2	4	12.9850	77.5533	560010

	Output	Feedback	Unnamed: 12
0	Yes	1	Yes
1	Yes	1	Yes
2	Yes	0	Yes
3	Yes	1	Yes
4	Yes	1	Yes

Figure 3.8: All the customers who ordered the food again.

	Age	Gender	Marital Status	Occupation	Monthly Income
36	25	1	1	1	0
71	24	0	1	1	25000
88	25	1	1	1	0
89	28	1	2	3	25000
90	27	0	0	2	5000

	Educational Qualifications	Family size	latitude	longitude	Pin code
36	2	5	12.9770	77.5773	560009
71	2	3	12.9335	77.5691	560028
88	2	5	12.9770	77.5773	560009
89	1	2	13.0289	77.5400	560022
90	2	5	13.0289	77.5400	560022

	Output	Feedback	Unnamed: 12
36	No	1	No
71	No	1	No
88	No	1	No
89	No	0	No
90	No	1	No

Figure 3.9: All the customers who not ordered the food again.

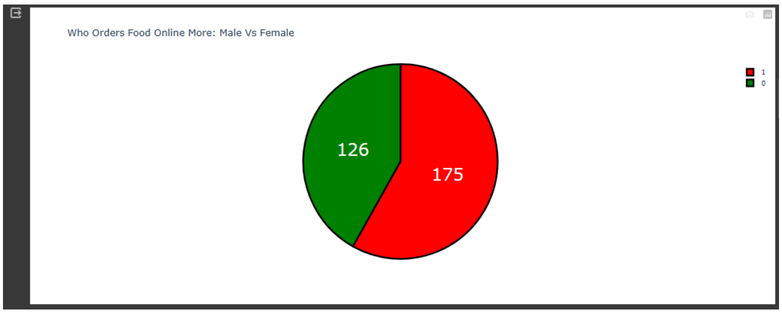

Figure 3.10: Find who orders food more online.

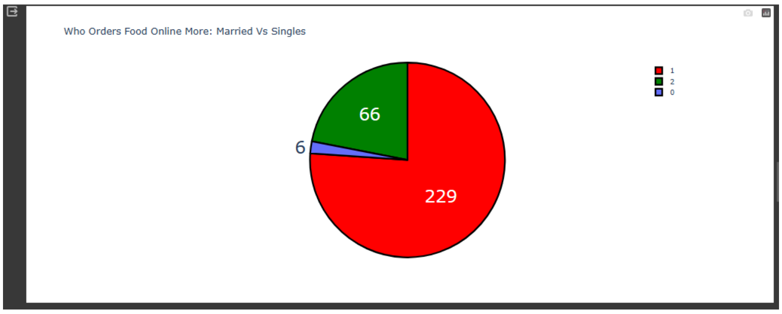

Figure 3.11: The marital status of the customers who ordered again.

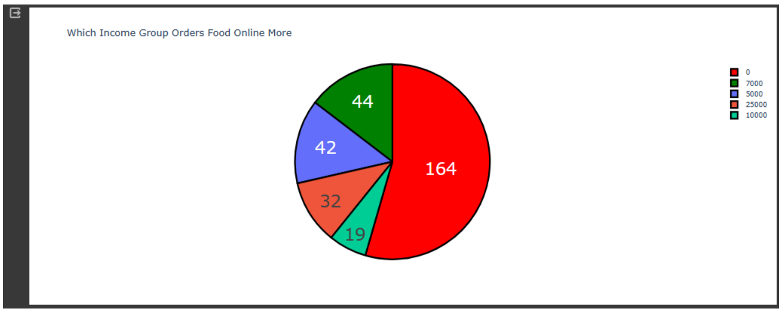

Figure 3.12: The income group of the customers who ordered the food again.

3.1.3 Key Findings and Observations

1. **Algorithmic Performance:**
    - **Random Forest Classifier:** Demonstrated robust performance in predicting online food ordering purchase. Its ensemble nature allowed for effective feature selection and handling of complex relationships in the data.
    - **Logistic Regression:** Showcased competitive performance, albeit generally trailing behind Random Forest. Its simplicity and interpretability made it a viable option for this prediction task.

- **Model Sensitivity:** Variations in test sizes, random state, or hyperparameters might significantly impact model performance across different algorithms.

2. **Impact of Features and Preprocessing:**

   - **Feature Importance:** Identified crucial features influencing purchase decisions, such as age, gender, income level, and feedback sentiment.
   - **Preprocessing Effectiveness:** Mapping categorical variables and scaling numerical values enhanced the models' predictive capabilities.

3. **Ensemble vs. Single Models:**

   - **Random Forest vs. Logistic Regression:** The ensemble approach of Random Forest outperformed Logistic Regression, suggesting the advantage of aggregating multiple models for this prediction task.

The results indicate Random Forest as a robust technique for predicting online food ordering purchase due to its ability to handle diverse data and capture complex decision boundaries. However, the study highlights the importance of feature selection and preprocessing in enhancing model performance. Future investigations could involve exploring advanced ensemble techniques or considering neural network architectures tailored for this specific prediction task.

Chapter 4

Conclusion

4.1 Discussion

The exploration of various data mining techniques for "Predicting online food ordering purchase" has revealed nuanced insights into their efficacy for this specific task. The Random Forest classifier showcased robustness, leveraging ensemble techniques to effectively capture intricate patterns in customer attributes and behaviors, resulting in superior predictive accuracy. Logistic Regression, while slightly trailing behind, provided valuable insights into the influence of specific customer attributes on the likelihood of placing online food orders, owing to its simplicity and interpretability. Sensitivity analysis underscored the necessity for precise parameter tuning and meticulous feature selection across different algorithms. Crucial features such as age, gender, income level, educational qualifications, and feedback sentiment significantly contributed to predictive power, emphasizing the importance of effective feature engineering. Future directions involve exploring advanced ensemble methods, expanding datasets, and enhancing model interpretability to further refine prediction accuracy in this domain. Overall, tailored algorithm selection, feature selection, and preprocessing emerged as crucial elements for accurate prediction of online food ordering behavior.

4.2 Limitations

There are several limitations are inherent in the process of predicting online food ordering purchases, impacting the accuracy and adaptability of the models utilized:

- **Data Volume and Complexity:** Dealing with extensive and complex datasets in the domain of online food ordering poses a challenge. The sheer volume of data and its multifaceted nature often leads to prolonged computation times and can strain computational resources, hindering swift analysis and model training.
- **Model Sensitivity to Parameters:** Similar to the broader scope of machine learning, models used in predicting food ordering habits are sensitive to hyperparameters.

- **Sparse and Imbalanced Data:** Online food ordering datasets might suffer from imbalanced class distributions, where certain categories or outcomes are significantly underrepresented. This imbalance can lead to biased models favoring the majority class and reduced accuracy in predicting minority classes.
- **Adaptability of Algorithms:** Fixed architectures within certain algorithms might limit their adaptability to varying complexities within online food ordering datasets. For instance, predefined layers in neural networks might not adequately capture evolving patterns or behaviors in customer preferences, potentially leading to reduced prediction accuracy.

These limitations highlight the complexities inherent in modeling the nuanced behaviors and preferences associated with online food ordering, emphasizing the need for continual refinement and adaptation of models to better capture the intricacies of this domain.

4.3 Scope of Future Work

There are several potential avenues for future development within this project. Some possible directions include:

- Behavioral Analytics Integration: Incorporate advanced behavioral analytics tools for deeper insights into customer preferences and interactions.
- Real-time Data Stream Analysis: Implement real-time analysis for agile model adaptation to changing customer behaviors.
- Enhanced Feature Selection: Explore novel features like user demographics and browsing history for improved model performance.
- Personalization and Contextual Models: Develop personalized models considering individual preferences and contextual information.

These future steps aim to address limitations, enhancing performance, scalability, and adaptability with larger datasets and better user interaction.

References

[1] MELIS NUR VERIR. (2023, January). online-food-order-prediction. Retrieved from https://www.kaggle.com/code/melisnurverir/online-food-order-prediction

[2] AshNumpy (Jul 30, 2023). ML-Online-Food-Ordering-Project. Retrieved from https://github.com/AshNumpy/ML-Online-Food-Ordering-Project/

[3] RACHEL MELTZER,(AUGUST 31, 2023). Retrieved from https://careerfoundry.com/en/blog/data-analytics/what-is-random-forest/.